How to generate HEPData submission using HEPData_lib

Jakub Kvapil

LA-UR-25-25779

What is HEPData_lib?

- HEPData_lib is the official tool by HEPData to generate its submissions
  - https://www.hepdata.net/submission
  - https://github.com/HEPData/hepdata_lib/ (repo with examples)
  - https://hepdata-lib.readthedocs.io/en/latest/usage.html (lib documentation)
- Continuously improved, easy to use, many example macros and documentation!
- **It can read directly ROOT objects** (TH1, TH2, TGraph, TGraphAsymmErrors) and parse TGraph.C and txt objects
- HEPData_lib manipulates python lists, if you already use pyplot for your plots, no ROOT handling is needed!
- It can be installed on SDCC or local using: python -m pip install hepdata_lib
- If you want to read ROOT files and are missing pyroot in your SDCC environment you can install both as:
  - conda config --set channel_priority strict
  - conda create -c conda-forge --name <my-environment> root hepdata-lib
  - conda activate <my-environment>

Using HEPData_lib – Initiate submission

```
from hepdata_lib import Submission, Table, Variable, Uncertainty, RootFileReader

# Load ROOT file using RootFileReader
reader = RootFileReader("spectrum_output.root")

# Initialize submission
submission = Submission()
```

imports

Open ROOT file with histograms

Initiate submission

Check the documentation and example codes for all the submission features!

submission.read_abstract("abstract.txt“) # add abstract

submission.add_link("arXiv“,“https link") # add arXiv link

submission.add_record_id(123456,"inspire“) # add inspire number

submission.add_additional_resource(“data file“,“data.root“, copy_file=True) # add file

Using HEPData_lib – Creating table with keywords

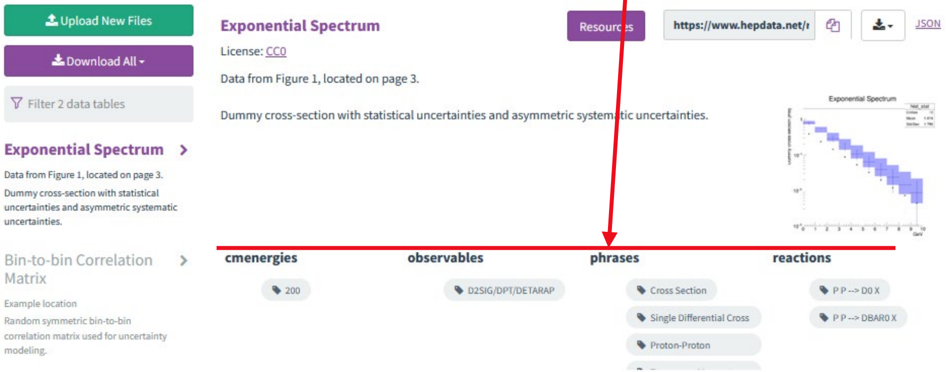
- Table holds one figure with multiple histograms that share axis – ideally one Table per Figure

```
# ------------------ Table 1: Spectrum -----------------------
#This objects hold page on HEPDATA - idealy 1 Table per figure
table1 = Table("Exponential Spectrum")
table1.description = "Dummy cross-section with statistical uncertainties and asymmetric systematic uncertainties."
table1.location = "Data from Figure 1, located on page 3."
# Add some keywords so it is searchable on HEPDATA
table1.keywords["observables"] = ["D2SIG/DPT/DETARAP"]
table1.keywords["cmenergies"] = ["200"]
table1.keywords["reactions"] = ["P P --> D0 X", "P P --> DBAR0 X"]
table1.keywords["phrases"] = ["Cross Section", "Single Differential Cross Section","Proton-Proton Scattering","Transverse Momentum Dependence"]
# Add image, can be pdf
table1.add_image("spectrum.png")
```

Do not forget to add keywords!
so people can easily search similar measurements
https://hepdata-submission.readthedocs.io/en/latest/keywords.html

Also add picture of the Figure!
This will automatically create a thumbnail

Tables



Using HEPData_lib – Reading the data from ROOT file

```python
# Get data from TH1D and TGraphAsymmErrors
hist = reader.read_hist_1d("hist_stat")
graph = reader.read_graph("graph_asym")

# This is the structure of the loaded objects (dictionary of lists)
#print(graph.keys())
#dict_keys(['x', 'y', 'dx', 'dy'])

# Define x-axis (independent)
histogram_x = Variable("$p_{T}$", is_independent=True, is_binned=False, units="GeV")
histogram_x.values = hist["x"]
# Define y-axis values (since we have 1D plot, this is dependent - ir depends on the independent variable x)
histogram_y = Variable("Cross-section", is_independent=False, is_binned=False, units="mb")
histogram_y.values = hist["y"]
#Define uncertainties, you have have any number you want, you are naming them: norm,lumi, etc.
#stat unc
stat_unc = Uncertainty("stat", is_symmetric=True)
stat_unc.values = hist["dy"]
#sys unc
sys_unc = Uncertainty("sys", is_symmetric=False)
sys_unc.values = graph["dy"]

#accidentelly saved symetrical errors in Assymerrors? No problem!
#sys_unc = Uncertainty("sys", is_symmetric=True)
#sys_unc.values =[high for (_, high) in graph["dy"]]

# Add uncertainty to data
histogram_y.add_uncertainty(stat_unc)
histogram_y.add_uncertainty(sys_unc)

# Add variables to the table
table1.add_variable(histogram_x)
table1.add_variable(histogram_y)

#Lets add another histogram in the same table, without any uncertainties
hist_halved = reader.read_hist_1d("hist_halved")
histogram_y_halved = Variable("Cross-section halved", is_independent=False, is_binned=False, units="mb")
histogram_y_halved.values = hist_halved["y"]
table1.add_variable(histogram_y_halved)
```

Read TH1D and TGraphAsymmErrors

Data is loaded into dictionary, you can easily validate them and manipulate

Define what is independent variable (the x-axis in 1D case)

- Is_binned specifies if you are providing bin centers (False) or bin edges (True)
- “True” supports also asymmetrical x axis bins

Define what the dependent variable (y axis) on the independent x

Take symmetrical uncertainties of the TH1D as statistical errors

Take asymmetrical uncertainties of the TGraph as systematical errors

- You can have as many unc. You want – global, lumi, norm etc.

Accidentally stored symmetrical uncertainties in TGraphAsymmError? No problem! Easy list operation

Appends the uncertainties to a corresponding variable

Add variables to a table

Let's add another histogram that shares x axis to the same Figure

Using HEPData_lib – Lets add another Table - TH2

```python
# ------------------ Table 2: Correlation Matrix ------------------
#lib also supports 2D hsitograms!
table2 = Table("Bin-to-bin Correlation Matrix")
table2.description = "Random symmetric bin-to-bin correlation matrix used for uncertainty modeling."
table2.add_image("corr_matrix.png")
corr_matrix = reader.read_hist_2d("corr_matrix")

# Create variable objects, since we have 2D histogram, x and y are independent variables
x = Variable("First Axis", is_independent=True, is_binned=True)
x.values = corr_matrix["x_edges"]
y = Variable("Second Axis", is_independent=True, is_binned=True)
y.values = corr_matrix["y_edges"]
correlation = Variable("Correlation coefficient", is_independent=False, is_binned=False)
correlation.values = corr_matrix["z"]
#Uncertainties can be added similarly to the 1D case

for var in [x,y,correlation]:
    table2.add_variable(var)
```

2D histograms can be also loaded!

Providing bin edges, so is_binned=True

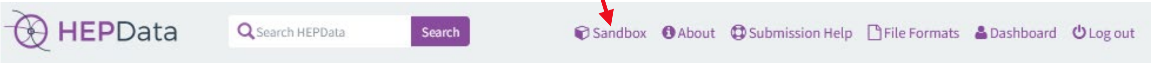
Now Add both Tables to the submission and generate the submission files as:

```python
# Add tables to submission
submission.add_table(table1)
submission.add_table(table2)

# Output submission to directory
submission.create_files("submission_hepdata",remove_old=True)
```

If there is some error in generated YAML it will tell you when you create_files

Now Log in to HEPData and upload the submission to Sandbox to see if it looks like you wanted!



Using HEPData_lib – Created submission - Table 1

Upload New Files

Download All

Filter 2 data tables

Exponential Spectrum

Data from Figure 1, located on page 3.

Dummy cross-section with statistical uncertainties and asymmetric systematic uncertainties.

Bin-to-bin Correlation Matrix

Example location

Random symmetric bin-to-bin correlation matrix used for uncertainty modeling.

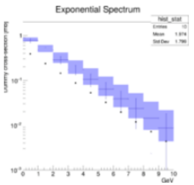
Exponential Spectrum

License: CC0

Data from Figure 1, located on page 3.

Dummy cross-section with statistical uncertainties and asymmetric systematic uncertainties.

Resources https://www.hepdata.net/ JSON



cmenergies

200

observables

D2SIG/DPT/DETARAP

phrases

Cross Section

Single Differential Cross Section

Proton-Proton Scattering

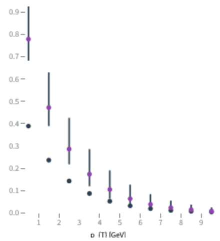
reactions

P P --> D0 X

P P --> DBAR0 X

p_T [GeV]	Cross-section [mb]	Cross-section halved [mb]
0.5	0.7788 ±0.08825 stat +0.11682 -0.03894 sys	0.3894
1.5	0.47237 ±0.068729 stat +0.14171 -0.047237 sys	0.23618
2.5	0.2865 ±0.053526 stat +0.12893 -0.042976 sys	0.14325
3.5	0.17377 ±0.041686 stat +0.10426 -0.034755 sys	0.086887
4.5	0.1054 ±0.032465 stat +0.079049 -0.02635 sys	0.0527
5.5	0.063928 ±0.025284 stat +0.057535 -0.019178 sys	0.031964
6.5	0.038774 ±0.019691 stat +0.040713 -0.013571 sys	0.019387
7.5	0.023518 ±0.015335 stat +0.028221 -0.0094071 sys	0.011759
8.5	0.014264 ±0.011943 stat +0.019257 -0.0064189 sys	0.0071321
9.5	0.0086517 ±0.0093014 stat +0.012978 -0.0043258 sys	0.0043258

Visualize



Sum errors Log Scale (X) Log Scale (Y)

Using HEPData_lib – Created submission - Table 2

Upload New Files

Download All

Filter 2 data tables

Exponential Spectrum

Data from Figure 1, located on page 3.

Dummy cross-section with statistical uncertainties and asymmetric systematic uncertainties.

Bin-to-bin Correlation Matrix

Example location

Random symmetric bin-to-bin correlation matrix used for uncertainty modeling.

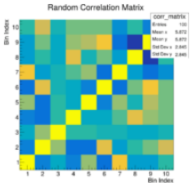
Bin-to-bin Correlation Matrix

License: CC0

Example location

Random symmetric bin-to-bin correlation matrix used for uncertainty modeling.

Resources https://www.hepdata.net/r JSON

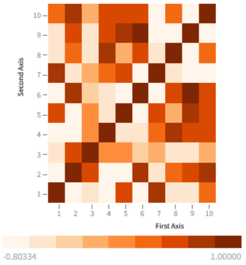


Showing 50 of 100 values

Show All 100 values

First Axis	Second Axis	Correlation coefficient
0.5 - 1.5	0.5 - 1.5	1.0
0.5 - 1.5	1.5 - 2.5	-0.37068
0.5 - 1.5	2.5 - 3.5	-0.13875
0.5 - 1.5	3.5 - 4.5	-0.35658
0.5 - 1.5	4.5 - 5.5	0.34342
0.5 - 1.5	5.5 - 6.5	-0.45626
0.5 - 1.5	6.5 - 7.5	0.65008
0.5 - 1.5	7.5 - 8.5	-0.21946
0.5 - 1.5	8.5 - 9.5	-0.2003
0.5 - 1.5	9.5 - 10.5	0.097076
1.5 - 2.5	0.5 - 1.5	-0.37068
1.5 - 2.5	1.5 - 2.5	1.0
1.5 - 2.5	2.5 - 3.5	0.288
1.5 - 2.5	3.5 - 4.5	-0.50662
1.5 - 2.5	4.5 - 5.5	-0.48634

Visualize



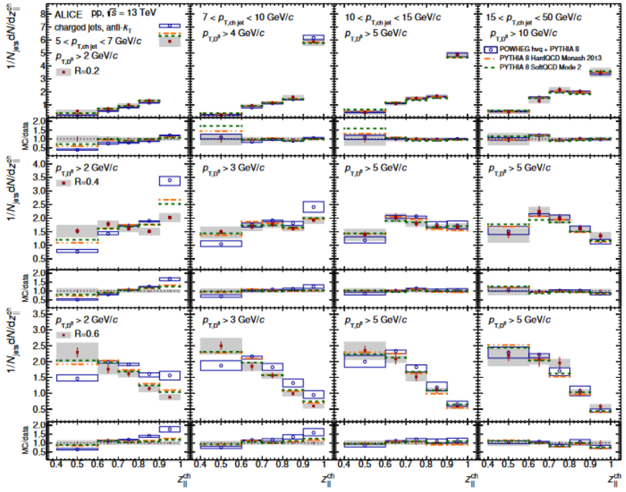
Brushing Enabled?

X Axis First Axis

Y Axis Second Axis

HEPData_lib supports very complex Figures!

$\sqrt{(s)}$	13 TeV											
Luminosity	25.81 ± 0.43 nb^{-1}											
Clustering algorithm	anti-k_T											
Recombination scheme	p_T											
R	0.2				0.4				0.6			
$\lvert\eta_{\mathrm{ch\ jet}}\rvert$	< 0.7				< 0.5				< 0.3			
$p_{T,\mathrm{ch\ jet}}$	$5-7$ GeV/c	$7-10$ GeV/c	$10-15$ GeV/c	$15-50$ GeV/c	$5-7$ GeV/c	$7-10$ GeV/c	$10-15$ GeV/c	$15-50$ GeV/c	$5-7$ GeV/c	$7-10$ GeV/c	$10-15$ GeV/c	$15-50$ GeV/c
p_{T,D^0}	$2-7$ GeV/c	$4-10$ GeV/c	$5-15$ GeV/c	$10-50$ GeV/c	$2-7$ GeV/c	$3-10$ GeV/c	$5-15$ GeV/c	$5-50$ GeV/c	$2-7$ GeV/c	$3-10$ GeV/c	$5-15$ GeV/c	$5-50$ GeV/c
$z_{\parallel}^{\mathrm{ch}}$	$1/N_{\mathrm{jets}} dN/dz_{\parallel}^{\mathrm{ch}}$											
0.4 - 0.6	0.51945 ±0.068089 stat +0.087268 -0.097137 sys	0.20871 ±0.043386 stat +0.058646 -0.06929 sys	0.40759 ±0.077445 stat +0.11779 -0.14551 sys	0.48996 ±0.093217 stat +0.1514 -0.16414 sys	1.5269 ±0.10523 stat +0.21072 -0.22599 sys	1.4924 ±0.099393 stat +0.18655 -0.22833 sys	1.4037 ±0.10528 stat +0.1909 -0.22319 sys	1.4125 ±0.14737 stat +0.20906 -0.30228 sys	2.2938 ±0.15294 stat +0.29361 -0.30508 sys	2.5011 ±0.13117 stat +0.2326 -0.26262 sys	2.3535 ±0.13892 stat +0.26359 -0.28948 sys	2.1864 ±0.17812 stat +0.24707 -0.31922 sys
0.6 - 0.7	0.7099 ±0.078096 stat +0.085188 -0.10081 sys	0.95137 ±0.089444 stat +0.086574 -0.10275 sys	1.1023 ±0.12904 stat +0.09921 -0.12456 sys	1.2999 ±0.18411 stat +0.14429 -0.15729 sys	1.7852 ±0.11118 stat +0.11782 -0.13211 sys	1.6925 ±0.11142 stat +0.14217 -0.15402 sys	2.0189 ±0.15283 stat +0.14132 -0.15546 sys	2.2414 ±0.21035 stat +0.16362 -0.17483 sys	1.762 ±0.13682 stat +0.14624 -0.15505 sys	1.8494 ±0.12935 stat +0.12946 -0.13686 sys	2.0818 ±0.17239 stat +0.099925 -0.11033 sys	2.0579 ±0.22692 stat +0.13582 -0.14405 sys
0.7 - 0.8	1.018 ±0.075462 stat +0.08246 -0.096712 sys	1.1593 ±0.084887 stat +0.090424 -0.1055 sys	1.5094 ±0.12816 stat +0.12377 -0.13434 sys	2.1979 ±0.1957 stat +0.14067 -0.15166 sys	1.6201 ±0.088144 stat +0.10855 -0.11827 sys	1.7695 ±0.094709 stat +0.10263 -0.11148 sys	1.8041 ±0.12837 stat +0.099226 -0.10825 sys	1.978 ±0.17774 stat +0.1523 -0.16022 sys	1.6203 ±0.1054 stat +0.097217 -0.1037 sys	1.5626 ±0.098353 stat +0.087507 -0.093757 sys	1.519 ±0.13303 stat +0.12152 -0.12759 sys	1.9581 ±0.1879 stat +0.11749 -0.12532 sys
0.8 - 0.9	1.3333 ±0.071417 stat +0.094662 -0.10666 sys	1.6037 ±0.089907 stat +0.13151 -0.14273 sys	1.6886 ±0.12329 stat +0.10132 -0.11483 sys	2.0046 ±0.17454 stat +0.12228 -0.13632 sys	1.5141 ±0.069213 stat +0.12567 -0.13172 sys	1.6223 ±0.082881 stat +0.079494 -0.087606 sys	1.693 ±0.11004 stat +0.059255 -0.081264 sys	1.6082 ±0.14679 stat +0.082018 -0.090058 sys	1.1508 ±0.074409 stat +0.10012 -0.10242 sys	0.98789 ±0.074119 stat +0.068164 -0.074092 sys	1.1019 ±0.10073 stat +0.060604 -0.067215 sys	1.0279 ±0.12978 stat +0.088397 -0.090453 sys
0.9 - 1.0	5.8999 ±0.11019 stat +0.34219 -0.38939 sys	5.8682 ±0.13431 stat +0.23473 -0.28754 sys	4.8844 ±0.1676 stat +0.20515 -0.24911 sys	3.5176 ±0.19471 stat +0.31658 -0.33417 sys	2.0267 ±0.062927 stat +0.14795 -0.15808 sys	1.931 ±0.073915 stat +0.11779 -0.12745 sys	1.6766 ±0.091214 stat +0.11904 -0.12574 sys	1.3473 ±0.11593 stat +0.12665 -0.128 sys	0.87933 ±0.053867 stat +0.083536 -0.085295 sys	0.59789 ±0.050273 stat +0.063376 -0.064572 sys	0.59039 ±0.064858 stat +0.067895 -0.068485 sys	0.58327 ±0.084454 stat +0.062993 -0.062993 sys



https://www.hepdata.net/record/134049

Los Alamos NATIONAL LABORATORY

Using HEPData_lib – example on sPHENIX/tutorials

- https://github.com/sPHENIX-Collaboration/tutorials/tree/master/HEPData